

(19)



JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 10269060 A

(43) Date of publication of application: 09.10.98

(51) Int. Cl

G06F 7/72

G09C 1/00

// G06F 17/10

(21) Application number: 10014250

(71) Applicant: TOSHIBA CORP

(22) Date of filing: 27.01.98

(72) Inventor: SHINPO ATSUSHI

(30) Priority: 27.01.97 JP 09 12667

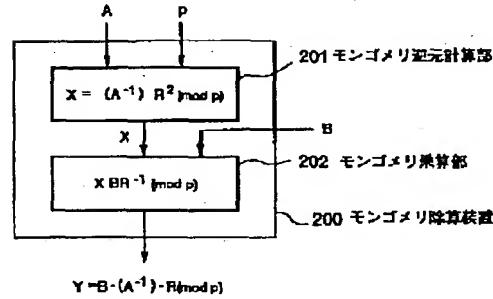
(54) MONTGOMERY DIVISION DEVICE,
MONTGOMERY INVERSE ELEMENT
CALCULATION DEVICE, MONTGOMERY
DIVISION METHOD AND MONTGOMERY
INVERSE ELEMENT CALCULATION METHOD

COPYRIGHT: (C)1998,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a Montgomery division device capable of obtaining a divided result in a Montgomery arithmetic area at a high speed.

SOLUTION: This Montgomery division device 200 for obtaining the divided result Y in the Montgomery arithmetic area to be $Y=B \cdot A' (-1) \cdot 2^n \bmod N$ for the integer (n) of $(n) \leq L$ when a bit length at the time of binarily expressing N is defined as L for a positive integer N, the positive integer A ($0 \leq A < N$ and A and N are mutually prime) and the positive integer B is provided with a Montgomery inverse element calculation part 201 for inputting the integer A and a modulus N and obtaining an inverse element $X=A (-1) \cdot 2^{(2n)} \bmod N$ and a Montgomery multiplication part 202 for inputting the obtained inverse element X, the modulus N and the B and obtaining the divided result $Y=B \cdot X \cdot 2^{-n} \bmod N$.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-269060

(43)公開日 平成10年(1998)10月9日

(51)Int.Cl.⁶
G 0 6 F 7/72
G 0 9 C 1/00
// G 0 6 F 17/10

識別記号
6 5 0

F I
G 0 6 F 7/72
G 0 9 C 1/00
G 0 6 F 15/31
6 5 0 A
Z

審査請求 未請求 請求項の数12 O L (全 18 頁)

(21)出願番号 特願平10-14250
(22)出願日 平成10年(1998)1月27日
(31)優先権主張番号 特願平9-12667
(32)優先日 平9(1997)1月27日
(33)優先権主張国 日本 (JP)

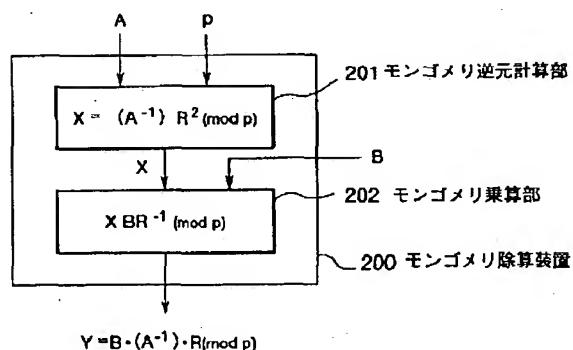
(71)出願人 000003078
株式会社東芝
神奈川県川崎市幸区堀川町72番地
(72)発明者 新保 淳
神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内
(74)代理人 弁理士 鈴江 武彦 (外6名)

(54)【発明の名称】 モンゴメリ除算装置及びモンゴメリ逆元計算装置並びにモンゴメリ除算方法及びモンゴメリ逆元計算方法

(57)【要約】

【課題】 本発明は、モンゴメリ演算域での除算結果を高速に求めることのできるモンゴメリ除算装置を提供すること。

【解決手段】 正の整数N、正の整数A ($0 \leq A < N$ 、AとNは互いに素)、正の整数Bについて、Nを2進表現したときのビット長をLとして、 $n \geq l$ なる整数nに対して、 $Y = B \cdot A^{-1} \cdot 2^{n} \bmod N$ なるモンゴメリ演算域での除算結果Yを求めるモンゴメリ除算装置であって、整数Aと法Nを入力として逆元 $X = A^{-1} \cdot 2^{n} \bmod N$ を求めるモンゴメリ逆元計算部と、求められた逆元Xと法NとBを入力として除算結果 $Y = B \cdot X \cdot 2^{n} \bmod N$ を求めるモンゴメリ乗算部とを備えたことを特徴とする。



【特許請求の範囲】

【請求項1】 正の整数N、正の整数A（ $0 \leq A < N$ 、AとNは互いに素）、正の整数Bについて、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $Y = B \cdot A^{-1} \cdot 2^n \bmod N$ なるモンゴメリ演算域での除算結果Yを求めるモンゴメリ除算装置であって、

整数Aと法Nを入力として逆元 $X = A^{-1} \cdot 2^{2n} \bmod N$ を求めるモンゴメリ逆元計算手段と、

求められた逆元Xと法NとBを入力として除算結果 $Y = B \cdot X \cdot 2^{-n} \bmod N$ を求めるモンゴメリ乗算手段とを備えたことを特徴とするモンゴメリ除算装置。

【請求項2】 前記モンゴメリ逆元計算手段は、

整数Aと法Nを入力として中間結果 $C = A^{-1} \cdot 2^k \bmod N$ とパラメータk（ $L \leq k \leq 2L$ ）を求める逆元計算手段と、

求められた中間結果Cとパラメータkと法Nを入力として逆元 $X = C \cdot 2^{2n-k} \bmod N$ を求める逆元補正手段とを有することを特徴とする請求項1に記載のモンゴメリ除算装置。

【請求項3】 正の整数N、正の整数A（ $0 \leq A < N$ 、AとNは互いに素）、正の整数Bについて、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $Y = B \cdot A^{-1} \cdot 2^n \bmod N$ なるモンゴメリ演算域での除算結果Yを求めるモンゴメリ除算装置であって、

整数Aと法Nを入力として第1の中間結果 $C = A^{-1} \cdot 2^k \bmod N$ とパラメータk（ $L \leq k \leq 2L$ ）を求める逆元計算手段と、

求められた第1の中間結果Cと法NとBを入力として第2の中間結果 $D = B \cdot C \cdot 2^{-n} \bmod N$ を求めるモンゴメリ乗算手段と、

このモンゴメリ乗算手段により求められた第2の中間結果Dと前記逆元計算手段により求められたパラメータkと法Nを入力として除算結果 $Y = D \cdot 2^{2n-k} \bmod N$ を求める逆元補正手段とを備えたことを特徴とするモンゴメリ除算装置。

【請求項4】 正の奇整数N、正の整数A（ $0 \leq A < N$ 、AとNは互いに素）について、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $X = A^{-1} \cdot 2^{2n} \bmod N$ なるモンゴメリ演算域での逆元Xを求めるモンゴメリ逆元計算装置であって、

整数Aと法Nを入力として中間結果 $C = A^{-1} \cdot 2^k \bmod N$ とパラメータk（ $L \leq k \leq 2L$ ）を求める逆元計算手段と、

求められた中間結果Cとパラメータkと法Nを入力として逆元 $X = C \cdot 2^{2n-k} \bmod N$ を求める逆元補正手段とを備えたことを特徴とするモンゴメリ逆元計算装置。

【請求項5】 内部に中間変数を記憶する複数のレジスタと、

前記レジスタを右または左にシフトするビットシフト器と、

2つのレジスタの内容の加算または減算を行う加減算器と、

2つのレジスタの内容の大小比較およびレジスタ内部の所定のビット位置の値の判定を行う判定器とを用いて前記逆元計算部および前記逆元補正部を構成することを特徴とする請求項4に記載のモンゴメリ逆元計算装置。

【請求項6】 正の奇整数N、正の整数A（ $0 \leq A < N$ 、AとNは互いに素）について、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $X = A^{-1} \cdot 2^{2n} \bmod N$ なるモンゴメリ演算域での逆元Xを求めるモンゴメリ逆元計算装置であって、

初期状態を2進表現にて $U = N$ 、 $V = A$ 、 $T = 0$ 、 $S = 1$ 、 $k = 0$ とし、

U の最下位ビットが0ならば、 U を右シフトし、 S を左シフトし、 k を1増加するとともに、 V の最下位ビットが0ならば、 V を右シフトし、 T を左シフトし、 k を1増加する処理と、

20 U の最下位ビットが1かつ V の最下位ビットが1で、 $U > V$ ならば、 U から V を減じ、 U を右シフトし、 T に S を加え、 S を左シフトし、 k を1増加する処理と、
 U の最下位ビットが1かつ V の最下位ビットが1で、 $U \leq V$ ならば、 V から U を減じ、 V を右シフトし、 S に T を加え、 T を左シフトし、 k を1増加する処理からなる一連のループ処理を、 $V > 0$ の間、繰り返し、
 $V = 0$ になった場合、 $T \geq N$ ならば T から N を減じた後に、 N から T を減じた値を T とし、 $T < N$ ならば N から T を減じた値を T とする逆元計算手段と、

初期状態を $i = 0$ とし、
前記逆元計算手段により求められた T を左シフトした後、 $T \geq N$ ならば T から N を減じて i を1増加し、 $T < N$ ならば i を1増加するループ処理を、 $i < 2n - k$ の間、繰り返し、
 $i = 2n - k$ になったときの T を逆元 X とする逆元補正手段とを備えたことを特徴とするモンゴメリ逆元計算装置。

【請求項7】 初期状態としてNが設定されるUレジスタと、

初期状態としてAが設定されるVレジスタと、
初期状態として0が設定されるTレジスタと、
初期状態として1が設定されるSレジスタと、
初期状態として0が設定されるkレジスタと、
Uレジスタの右シフト、Vレジスタの右シフト、Tレジスタの左シフト、およびSレジスタの左シフトのうち指定されたものを実行するビットシフト器と、

UレジスタからVレジスタの内容を減じる処理、VレジスタからUレジスタの内容を減じる処理、TレジスタにSレジスタの内容を加える処理、SレジスタにTレジスタの内容を加える処理、TレジスタからNを減じる処

理、NからTレジスタの内容を減じる処理、およびkレジスタに1を加える処理のうち指定されたものを実行する加減算器と、

Uレジスタの最下位ビットが0であるか否か、Vレジスタの最下位ビットが0であるか否か、Vレジスタの値が0になったか否か、およびTレジスタの値がN以上であるか否かを判断する判定器と、

前記ビットシフト器、前記加減算器および前記判定器を制御する制御部を備えたことを特徴とする請求項6に記載のモンゴメリ逆元計算装置。

【請求項8】正の整数N、正の整数A（ $0 \leq A < N$ 、AとNは互いに素）、正の整数Bについて、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $Y = B \cdot A^{-1} \cdot 2^n \bmod N$ なるモンゴメリ演算域での除算結果Yを求めるモンゴメリ除算方法であって、

整数Aと法Nを入力として逆元 $X = A^{-1} \cdot 2^{2n} \bmod N$ を求める第1のステップと、

求められた逆元Xと法NとBを入力として除算結果 $Y = B \cdot X \cdot 2^{-n} \bmod N$ を求める第2のステップとを有することを特徴とするモンゴメリ除算方法。

【請求項9】前記第1のステップは、

整数Aと法Nを入力として中間結果 $C = A^{-1} \cdot 2^k \bmod N$ とパラメータk（ $L \leq k \leq 2L$ ）を求め、求められた中間結果Cとパラメータkと法Nを入力として逆元 $X = C \cdot 2^{2n-k} \bmod N$ を求めるものであることを特徴とする請求項8に記載のモンゴメリ除算方法。

【請求項10】正の整数N、正の整数A（ $0 \leq A < N$ 、AとNは互いに素）、正の整数Bについて、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $Y = B \cdot A^{-1} \cdot 2^n \bmod N$ なるモンゴメリ演算域での除算結果Yを求めるモンゴメリ除算方法であって、

整数Aと法Nを入力として第1の中間結果 $C = A^{-1} \cdot 2^k \bmod N$ とパラメータk（ $L \leq k \leq 2L$ ）を求める第1のステップと、

この第1のステップにより求められた第1の中間結果Cと法NとBを入力として第2の中間結果 $D = B \cdot C \cdot 2^{-n} \bmod N$ を求める第2のステップと、

この第2のステップにより求められた第2の中間結果Dと前記第1のステップにより求められたパラメータkと法Nを入力として除算結果 $Y = D \cdot 2^{2n-k} \bmod N$ を求める第3のステップとを有することを特徴とするモンゴメリ除算方法。

【請求項11】正の奇整数N、正の整数A（ $0 \leq A < N$ 、AとNは互いに素）について、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $X = A^{-1} \cdot 2^{2n} \bmod N$ なるモンゴメリ演算域での逆元Xを求めるモンゴメリ逆元計算方法であって、

整数Aと法Nを入力として中間結果 $C = A^{-1} \cdot 2^k \bmod N$ と

o d Nとパラメータk（ $L \leq k \leq 2L$ ）を求め、求められた中間結果Cとパラメータkと法Nを入力として逆元 $X = C \cdot 2^{2n-k} \bmod N$ を求める特徴とするモンゴメリ逆元計算方法。

【請求項12】正の奇整数N、正の整数A（ $0 \leq A < N$ 、AとNは互いに素）について、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $X = A^{-1} \cdot 2^{2n} \bmod N$ なるモンゴメリ演算域での逆元Xを求めるモンゴメリ逆元計算方法であって、

10 初期状態を2進表現にてU=N、V=A、T=0、S=1、k=0とし、Uの最下位ビットが0ならば、Uを右シフトし、Sを左シフトし、kを1増加するとともに、Vの最下位ビットが0ならば、Vを右シフトし、Tを左シフトし、kを1増加する処理と、

Uの最下位ビットが1かつVの最下位ビットが1で、U>Vならば、UからVを減じ、Uを右シフトし、TにSを加え、Sを左シフトし、kを1増加する処理と、Uの最下位ビットが1かつVの最下位ビットが1で、U≤Vならば、VからUを減じ、Vを右シフトし、SにTを加え、Tを左シフトし、kを1増加する処理からなる一連のループ処理を、V>0の間、繰り返し、V=0になった場合、T≥NならばTからNを減じた後に、NからTを減じた値をTとし、T<NならばNからTを減じた値をTとする第1のステップと、

初期状態をi=0とし、前記第1のステップにより求められたTを左シフトした後、T≥NならばTからNを減じてiを1増加し、T<Nならばiを1増加するループ処理を、i<2n-kの間、繰り返し、

40 i=2n-kになったときのTを逆元Xとする第2のステップとを有することを特徴とするモンゴメリ逆元計算方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、計算機ネットワークでのデータ通信におけるデータの暗号化や通信相手の確認に利用される公開鍵暗号など奇数の整数を法とする多倍長の四則演算を繰り返し利用する処理のためのモンゴメリ除算装置及びモンゴメリ逆元計算装置並びにモンゴメリ除算方法及びモンゴメリ逆元計算方法に関する。

【0002】

【従来の技術】情報通信ネットワークや計算機システムでは、電子的な情報の交換・蓄積が行われる。そのようなシステムが、大規模化し不特定多数のユーザが利用する状況では、悪意のユーザによる情報の盗聴や改ざなどがある問題となり、その対策として公開鍵暗号技術を利用する場合が多い。

50 【0003】公開鍵暗号では、多倍長の奇整数を法とする演算で実現される方が多く、その高速化が性能に影

響を与える。多倍長の奇整数を法とする四則演算の中では、特に乗算と除算が処理時間に与える影響が大きい。このうち、乗算を繰り返し実行する場合に適した計算アルゴリズムとして、モンゴメリ算法が知られている。

モンゴメリ算法は文献(1) P. L. Montgomery, "Modular multiplication without trial division", Math. of Comp., Vol. 44, No. 170, pp. 519-521 (1985) に詳しい。

【0004】モンゴメリ算法は多倍長の剩余乗算を多倍長乗算2回程度の処理量で計算する方法である。多倍長の剩余算は多倍長の乗算よりも性能が劣ることが多く、その分の高速化が実現できる。このモンゴメリ算法はモンゴメリ演算域の元(これも同じ剩余系である)の乗算アルゴリズムであり、一般の剩余系での乗算をするには、まず乗数と被乗数をモンゴメリ演算域に変換し、次にモンゴメリ乗算を行ない、最後にモンゴメリ演算域から元の剩余系に結果を逆変換する。モンゴメリ変換とモンゴメリ逆変換はいずれも多倍長乗算1回程度の処理であるため、剩余乗算を繰り返し行なう、べき乗演算では変換と逆変換のオーバーヘッドが少なく、高速化が可能である。したがって、RSA(Rivest-Shamir-Adleman)暗号など多くの公開鍵暗号では、剩余系でのべき乗剩余演算 $c = m^e \bmod N$ をその基本演算としているため、このモンゴメリ算法を有効に利用することができる(ただし、単純にいくつかの乗算を行なうだけの場合には、変換と逆変換のオーバーヘッドのため必ずしも効率化にはつながらない)。

【0005】ところで、近年、新しい暗号方式が種々研究・提案されており、例えば楕円曲線暗号が公開鍵暗号の中で注目を集めている。これは、楕円曲線上の離散対数問題がRSA暗号のベースとなっている合成数の素因数分解に比べて計算量的に困難であるという予想に基づいている。

【0006】ここで、楕円曲線暗号の基本演算について簡単に説明する。

【0007】有限体 F_p (ただし、 $p > 3$)において、 $E(a, b) / F_p : y^2 = x^3 + ax + b \bmod p$

ただし、 $0 \leq a, b < p$ なる整数、 $4a^3 + 27b^2 \neq 0 \bmod p$ で定義される曲線を有限体 F_p 上の楕円曲線といふ。楕円曲線上の点とは、上式を満たす (x, y) の組(ただし、 $0 \leq x, y < p$ なる整数)に無限遠点 ∞ を加えたものをいう。この無限遠点 ∞ は加算に関する単位元となる。

【0008】楕円曲線上の点は以下に示す加算に関して群をなす。楕円曲線上の点 $P = (x_1, y_1), Q = (x_2, y_2)$ の加算点を $S(x_3, y_3)$ とすると、次のようになる。ただし、 $-P = (x_1, -y_1)$ である。

【0009】(1) Q が単位元 ∞ のとき、

$$S = P + Q = Q + P = P$$

(2) $Q = -P$ のとき、

$$S = P + Q = Q + P = \infty$$

(3) $P \neq Q$ のとき(ただし上記(1)(2)以外)、

$$x_3 = (y_2 - y_1)^2 / (x_2 - x_1)^2 - x_1 - x_2 \bmod p$$

$$y_3 = (y_2 - y_1)(x_1 - x_3) / (x_2 - x_1) - y_1 \bmod p$$

(4) $P = Q$ のとき、

$\cdot y_1 \neq 0$ の場合

$$x_3 = (3x_1^2 + a)^2 / (2y_1)^2 - 2x_1 \bmod p$$

$$y_3 = (3x_1^2 + a)(x_1 - x_3) / (2y_1) - y_1 \bmod p$$

$\cdot y_1 = 0$ の場合

$$S = \infty$$

また、楕円曲線上の点 $P = (x_1, y_1)$ の e (整数)倍演算は上記加算の繰り返しとして次のように定義される。

$$eP = P + P + \dots + P \quad (P \text{を } e \text{回加算する})$$

ただし、 $e < 0$ の場合は、点 $(-P)$ を $(-e)$ 倍する($(-e)$ は正である)。 $e = 0$ のときには $0P = \infty$ とする。

【0010】楕円曲線暗号では、この楕円曲線上の点のスカラー倍演算(ベキ加算)が基本演算となる。例えば、楕円ElGamal暗号、楕円ElGamal署名、楕円DHなどにおける処理の大半を占める演算である。

【0011】したがって、RSA暗号が剩余乗算を基本演算としているのに対して、楕円曲線暗号では、基本演算を実現するのに四則演算が必要となる。

【0012】さて、楕円曲線暗号のように基本演算が多倍長の四則演算の繰り返し処理であるような場合、四則演算の中で処理に時間を要するものは、剩余乗算および剩余除算であり、暗号処理全体を高速化するには、これら剩余乗算および剩余除算を高速化する必要がある。このうち前者の剩余乗算は、文献(1)のモンゴメリ乗算のアルゴリズムなどを用いれば良い。

【0013】一方、剩余除算は逆元計算と剩余乗算との組合せで実現でき、一般に逆元は拡張ユークリッド互除法と呼ばれる算法で計算できる。しかし、一般にこのアルゴリズムはそれほど高速ではない。より高速な逆元の計算法として、多倍長整数の右シフト(1/2倍)、加算、減算で構成された計算法が、例えば文献(2) Kalliski, B. S. Jr., "The Montgomery inverse and its application", IEEE Trans. Comp., Vol. 44, No. 8, pp. 1064-1065, (Aug. 1995) に示されている。

【0014】しかしながら、前述の文献(1)のモンゴメリ乗算と文献(2)の剩余系での除算の高速計算法をそのまま適用することはできない。なぜなら、モンゴメリ演算域と元の剩余系との変換・逆変換を乗算や除算のたびに実行しなければならず、オーバーヘッドが大きくなるからである。

【0015】また、剩余除算をモンゴメリ演算域で効率良く求めるアルゴリズムはなかった。

【0016】このように、剩余除算を高速化することは困難であるという問題点があった。

【0017】したがって、楕円曲線暗号のように基本演算が四則演算(剩余系演算)の繰り返し処理であるような暗号の処理を高速化することは困難であるという問題点があった。

【0018】

【発明が解決しようとする課題】以上説明したように、従来、公開鍵暗号の基本演算である剩余系での乗算と除算を含む演算の繰り返し処理を効率化する算法は実現されておらず、公開鍵暗号の一種である楕円曲線暗号などにおける全体としての処理時間の効率化が困難であるという問題点があった。

【0019】本発明は、上記事情を考慮してなされたもので、モンゴメリ演算域での逆元を高速に求めることのできるモンゴメリ逆元計算装置及び方法、モンゴメリ演算域での除算結果を高速に求めることのできるモンゴメリ除算装置及び方法を提供することを目的とする。

【0020】

【課題を解決するための手段】本発明(請求項1)は、正の整数N、正の整数A($0 \leq A < N$ 、AとNは互いに素)、正の整数Bについて、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $Y = B \cdot A^{-1} \cdot 2^n \bmod N$ なるモンゴメリ演算域での除算結果Yを求めるモンゴメリ除算装置であって、整数Aと法Nを入力として逆元X = $A^{-1} \cdot 2^{2n} \bmod N$ を求めるモンゴメリ逆元計算手段と、求められた逆元Xと法NとBを入力として除算結果Y = $B \cdot X \cdot 2^n \bmod N$ を求めるモンゴメリ乗算手段とを備えたことを特徴とする。

【0021】本発明(請求項2)は、請求項1に記載のモンゴメリ除算装置において、前記モンゴメリ逆元計算手段は、整数Aと法Nを入力として中間結果C = $A^{-1} \cdot 2^k \bmod N$ とパラメータk($L \leq k \leq 2L$)を求める逆元計算手段と、求められた中間結果Cとパラメータkと法Nを入力として逆元X = $C \cdot 2^{2n-k} \bmod N$ を求める逆元補正手段とを有することを特徴とする。

【0022】本発明(請求項3)は、正の整数N、正の整数A($0 \leq A < N$ 、AとNは互いに素)、正の整数Bについて、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $Y = B \cdot A^{-1} \cdot 2^n \bmod N$ なるモンゴメリ演算域での除算結果Yを求め

10

20

30

40

50

るモンゴメリ除算装置であって、整数Aと法Nを入力として第1の中間結果C = $A^{-1} \cdot 2^k \bmod N$ とパラメータk($L \leq k \leq 2L$)を求める逆元計算手段と、求められた第1の中間結果Cと法NとBを入力として第2の中間結果D = $B \cdot C \cdot 2^{-k} \bmod N$ を求めるモンゴメリ乗算手段と、このモンゴメリ乗算手段により求められた第2の中間結果Dと前記逆元計算手段により求められたパラメータkと法Nを入力として除算結果Y = $D \cdot 2^{2n-k} \bmod N$ を求める逆元補正手段とを備えたことを特徴とする。

【0023】本発明(請求項4)は、正の奇整数N、正の整数A($0 \leq A < N$ 、AとNは互いに素)について、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $X = A^{-1} \cdot 2^{2n} \bmod N$ なるモンゴメリ演算域での逆元Xを求めるモンゴメリ逆元計算装置であって、整数Aと法Nを入力として中間結果C = $A^{-1} \cdot 2^k \bmod N$ とパラメータk($L \leq k \leq 2L$)を求める逆元計算手段と、求められた中間結果Cとパラメータkと法Nを入力として逆元X = $C \cdot 2^{2n-k} \bmod N$ を求める逆元補正手段とを備えたことを特徴とする。

【0024】本発明(請求項5)は、請求項4に記載のモンゴメリ逆元計算装置において、内部に中間変数を記憶する複数のレジスタと、前記レジスタを右または左にシフトするビットシフト器と、2つのレジスタの内容の加算または減算を行う加減算器と、2つのレジスタの内容の大小比較およびレジスタ内部の所定のビット位置の値の判定を行う判定器とを用いて前記逆元計算部および前記逆元補正部を構成することを特徴とする。

【0025】本発明(請求項6)は、正の奇整数N、正の整数A($0 \leq A < N$ 、AとNは互いに素)について、Nを2進表現したときのビット長をLとして、 $n \geq L$ なる整数nに対して、 $X = A^{-1} \cdot 2^{2n} \bmod N$ なるモンゴメリ演算域での逆元Xを求めるモンゴメリ逆元計算装置であって、初期状態を2進表現にてU = N、V = A、T = 0、S = 1、k = 0とし、Uの最下位ビットが0ならば、Uを右シフトし、Sを左シフトし、kを1増加するとともに、Vの最下位ビットが0ならば、Vを右シフトし、Tを左シフトし、kを1増加する処理と、Uの最下位ビットが1かつVの最下位ビットが1で、U > Vならば、UからVを減じ、Uを右シフトし、TにSを加え、Sを左シフトし、kを1増加する処理と、Uの最下位ビットが1かつVの最下位ビットが1で、U ≤ Vならば、VからUを減じ、Vを右シフトし、SにTを加え、Tを左シフトし、kを1増加する処理からなる一連のループ処理を、V > 0の間、繰り返し、V = 0になった場合、T ≥ NならばTからNを減じた後に、NからTを減じた値をTとし、T < NならばNからTを減じた値をTとする逆元計算手段と、初期状態をi = 0とし、前記逆元計算手段により求められたTを左シフトした後、

$T \geq N$ ならば T から N を減じて i を 1 増加し、 $T < N$ ならば i を 1 増加するループ処理を、 $i < 2n - k$ の間、繰り返し、 $i = 2n - k$ になったときの T を逆元 X とする逆元補正手段とを備えたことを特徴とする。

【0026】本発明（請求項7）は、請求項6に記載のモンゴメリ逆元計算装置において、初期状態として N が設定される U レジスタと、初期状態として A が設定される V レジスタと、初期状態として 0 が設定される T レジスタと、初期状態として 1 が設定される S レジスタと、初期状態として 0 が設定される k レジスタと、 U レジスタの右シフト、 V レジスタの右シフト、 T レジスタの左シフト、および S レジスタの左シフトのうち指定されたものを実行するビットシフト器と、 U レジスタから V レジスタの内容を減じる処理、 V レジスタから U レジスタの内容を減じる処理、 T レジスタに S レジスタの内容を加える処理、 S レジスタに T レジスタの内容を加える処理、 T レジスタから N を減じる処理、 N から T レジスタの内容を減じる処理、および k レジスタに 1 を加える処理のうち指定されたものを実行する加減算器と、 U レジスタの最下位ビットが 0 であるか否か、 V レジスタの最下位ビットが 0 であるか否か、 V レジスタの値が 0 になったか否か、および T レジスタの値が N 以上であるか否かを判断する判定器と、前記ビットシフト器、前記加減算器および前記判定器を制御する制御部を備えたことを特徴とする。

【0027】本発明（請求項8）は、正の整数 N 、正の整数 A ($0 \leq A < N$ 、 A と N は互いに素)、正の整数 B について、 N を 2 進表現したときのビット長を L として、 $n \geq L$ なる整数 n に対して、 $Y = B \cdot A^{-1} \cdot 2^n \bmod N$ なるモンゴメリ演算域での除算結果 Y を求めるモンゴメリ除算方法であって、整数 A と法 N を入力として逆元 $X = A^{-1} \cdot 2^n \bmod N$ を求める第1のステップと、求められた逆元 X と法 N と B を入力として除算結果 $Y = B \cdot X \cdot 2^n \bmod N$ を求める第2のステップとを有することを特徴とする。

【0028】本発明（請求項9）は、請求項8に記載のモンゴメリ除算方法において、前記第1のステップは、整数 A と法 N を入力として中間結果 $C = A^{-1} \cdot 2^k \bmod N$ とパラメータ k ($L \leq k \leq 2L$) を求め、求められた中間結果 C とパラメータ k と法 N を入力として逆元 $X = C \cdot 2^{n-k} \bmod N$ を求めるものであることを特徴とする。

【0029】本発明（請求項10）は、正の整数 N 、正の整数 A ($0 \leq A < N$ 、 A と N は互いに素)、正の整数 B について、 N を 2 進表現したときのビット長を L として、 $n \geq L$ なる整数 n に対して、 $Y = B \cdot A^{-1} \cdot 2^n \bmod N$ なるモンゴメリ演算域での除算結果 Y を求めるモンゴメリ除算方法であって、整数 A と法 N を入力として第1の中間結果 $C = A^{-1} \cdot 2^k \bmod N$ とパラメータ k ($L \leq k \leq 2L$) を求める第1のステップと、

この第1のステップにより求められた第1の中間結果 C と法 N と B を入力として第2の中間結果 $D = B \cdot C \cdot 2^{-n} \bmod N$ を求める第2のステップと、この第2のステップにより求められた第2の中間結果 D と前記第1のステップにより求められたパラメータ k と法 N を入力として除算結果 $Y = D \cdot 2^{2n-k} \bmod N$ を求める第3のステップとを有することを特徴とする。

【0030】本発明（請求項11）は、正の奇整数 N 、正の整数 A ($0 \leq A < N$ 、 A と N は互いに素) について、 N を 2 進表現したときのビット長を L として、 $n \geq L$ なる整数 n に対して、 $X = A^{-1} \cdot 2^{2n} \bmod N$ なるモンゴメリ演算域での逆元 X を求めるモンゴメリ逆元計算方法であって、整数 A と法 N を入力として中間結果 $C = A^{-1} \cdot 2^k \bmod N$ とパラメータ k ($L \leq k \leq 2L$) を求め、求められた中間結果 C とパラメータ k と法 N を入力として逆元 $X = C \cdot 2^{2n-k} \bmod N$ を求ることを特徴とする。

【0031】本発明（請求項12）は、正の奇整数 N 、正の整数 A ($0 \leq A < N$ 、 A と N は互いに素) について、 N を 2 進表現したときのビット長を L として、 $n \geq L$ なる整数 n に対して、 $X = A^{-1} \cdot 2^{2n} \bmod N$ なるモンゴメリ演算域での逆元 X を求めるモンゴメリ逆元計算方法であって、初期状態を 2 進表現にて $U = N$ 、 $V = A$ 、 $T = 0$ 、 $S = 1$ 、 $k = 0$ とし、 U の最下位ビットが 0 ならば、 U を右シフトし、 S を左シフトし、 k を 1 増加するとともに、 V の最下位ビットが 0 ならば、 V を右シフトし、 T を左シフトし、 k を 1 増加する処理と、 U の最下位ビットが 1 かつ V の最下位ビットが 1 で、 $U > V$ ならば、 U から V を減じ、 U を右シフトし、 T に S を加え、 S を左シフトし、 k を 1 増加する処理と、 U の最下位ビットが 1 かつ V の最下位ビットが 1 で、 $U \leq V$ ならば、 V から U を減じ、 V を右シフトし、 S に T を加え、 T を左シフトし、 k を 1 増加する処理からなる一連のループ処理を、 $V > 0$ の間、繰り返し、 $V = 0$ になった場合、 $T \geq N$ ならば T から N を減じた後に、 N から T を減じた値を T とし、 $T < N$ ならば N から T を減じた値を T とする第1のステップと、初期状態を $i = 0$ とし、前記第1のステップにより求められた T を左シフトした後、 $T \geq N$ ならば T から N を減じて i を 1 増加し、 $T < N$ ならば i を 1 増加するループ処理を、 $i < 2n - k$ の間、繰り返し、 $i = 2n - k$ になったときの T を逆元 X とする第2のステップとを有することを特徴とする。

【0032】本発明によれば、モンゴメリ演算域のままでモンゴメリ逆元計算を行なうモンゴメリ逆元計算手段とモンゴメリ乗算手段を用いるため、モンゴメリ演算域の元を入力として、モンゴメリ演算域での除算結果を直接求めることができる。この結果、モンゴメリ演算域と元の剰余系との変換・逆変換のオーバーヘッドがないため、モンゴメリ演算域での除算が高速に実現できる。

【0033】また、本発明によれば、モンゴメリ演算域

のままモンゴメリ逆元計算を行なうことができ、モンゴメリ演算域と元の剰余系との変換・逆変換のオーバーヘッドがないため、モンゴメリ演算域での逆元計算が高速に実現できる。

【0034】また、本発明によれば、モンゴメリ演算域での逆元計算が多倍長レジスタの加減算とビットシフトで実現できるため、ソフトウェア実装・ハードウェア実装のどちらでも高速な装置構成が可能となる。さらに、モンゴメリ演算域での除算も高速な装置構成が実現できる。

【0035】したがって、楕円曲線暗号などのように剰余系での乗算と除算を含む演算の繰り返し処理を基本演算とする暗号において、全体としての処理時間を高速化することができる。

【0036】なお、装置に係る本発明は方法に係る発明としても成立し、方法に係る本発明は装置に係る発明としても成立する。

【0037】また、装置または方法に係る本発明は、コンピュータに当該発明に相当する手順を実行させるための（あるいはコンピュータを当該発明に相当する手段として機能させるための、あるいはコンピュータに当該発明に相当する機能を実現させるための）プログラムを記録したコンピュータ読み取り可能な記録媒体としても成立する。

【0038】

【発明の実施の形態】以下、図面を参照しながら発明の実施の形態を説明する。

【0039】なお、以下では、整数 s, u について $s \bmod u$ は、 s を u で割ったときの剰余を表すものとする。

【0040】前述したようにモンゴメリ演算域での乗算は一般に、剰余系での乗算よりも効率良く実装できることが知られており、RSA暗号など剰余系の乗算を繰り返す方式を実現する場合、モンゴメリ演算が利用されることがある。しかし、RSA暗号のように暗号化・復号化に乗算を用いるだけではなく、除算や逆元計算を用いる暗号もある。例えば、楕円曲線暗号では、剰余系での四則演算を組合せて単位演算が構成され、その単位演算を所定の回数繰り返すことで暗号アルゴリズムが構成される。ところが、モンゴメリ演算域での除算や逆元計算としては効率良く実装できるものがなく、楕円曲線暗号等のように剰余系での四則演算を組合せて単位演算が構成されるものについてはモンゴメリ演算域を利用した処理の効率化が困難であった。

【0041】本実施形態に係る除算装置、逆元計算装置は、それぞれモンゴメリ演算域での剰余除算、逆元計算の結果を効率的に求めることができるようにしたものである。

【0042】最初に、通常の剰余系 Z_p (p を法とする剰余系を表す) とモンゴメリ演算域との関係およびモン

ゴメリ演算域での演算について説明する。

【0043】図12に、モンゴメリ演算域と剰余系 Z_p の2つの代数系の間における、定義域、元、逆元、および加算、減算、乗算、除算の四則演算の関係を示す。

【0044】なお、法の値 p は整数であるが、好ましくは奇整数を用いる。また、暗号システムに適用する場合、 p には素数を用いることが多い。

【0045】剰余系 Z_p での元を a とするとき、それに対応するモンゴメリ演算域の元 A には、

$$10 \quad A = a \cdot R \bmod p$$

により与えられる。

【0046】一方、モンゴメリ演算域の元 A から剰余系 Z_p の元 a への逆変換は、

$$a = A \cdot R^{-1} \bmod p$$

により与えられる。

【0047】ここで、 R はモンゴメリ演算域を定義するパラメータであり、その条件は、

(i) R と法 p とは互いに素であること

(ii) $R > p$

の2つである。

【0048】一般に R は2のべき乗 ($R = 2^n$) とし、さらにハードウェアのワード長の倍数を用いることが多い。演算幅をできるだけ小さくするためには、法 p のビット数を L とした場合に、 $n \geq L$ を満たし、ワード長の倍数である最小の値を用いることが好ましい。以下では、簡単のために、 n は法 p のビット長 L と等しいものとして説明する。

【0049】図13(a)に、 $p = 23, R = 2^5$ ($n = 5$) とした場合の剰余系 Z_p の元 a とモンゴメリ演算域の元 A との対応関係を一例として示す(ただし、現実の暗号装置等で使用する p は非常に大きな整数である)。

【0050】モンゴメリ演算域での除算装置や逆元計算装置では、 $R = 2^n$ を用いるものとする。

【0051】次に、モンゴメリ演算域での逆元計算と四則演算は、剰余系 Z_p の元 a に対し $A = a \cdot R \bmod p$ なるモンゴメリ演算域の元が対応することを考慮して、図12のように定義できる。

【0052】まず、モンゴメリ演算域の加算および減算は、剰余系でのそれらと同様、

$$A + B \bmod p$$

$$A - B \bmod p$$

で定義される。

【0053】次に、モンゴメリ乗算は、

$$A \cdot B \cdot R^{-1} \bmod p$$

により与えられる。

【0054】他のモンゴメリ演算域での演算はモンゴメリ乗算 $A \cdot B \cdot R^{-1} \bmod p$ をもとにすると以下のようない定義される。

【0055】 A の逆元 X は、 A と X を乗数・被乗数とし

てモンゴメリ乗算を行なったときに、モンゴメリ演算域での単位元となるRを結果として与えるような値となる。すなわち、逆元Xは、

$$A \cdot X = R^2 \pmod{p}$$

を満たす値である。

【0056】したがって、モンゴメリ演算域での法をpとするAについての逆元計算は、

$$X = A^{-1} \cdot R^2 \pmod{p}$$

により与えられる。

【0057】後述する本実施形態に係る逆元計算装置は、入力Aと法pに対して、 $X = A^{-1} \cdot R^2 \pmod{p}$ を効率的に求める装置である。

【0058】同様にして、法をp、Aを除数、Bを被除数とするモンゴメリ除算は、

$$Y = B \cdot A^{-1} \cdot R \pmod{p}$$

$$= B \cdot A' \cdot R^{-1} \pmod{p} \quad (\text{ただし、} A' \text{ はモンゴメリ演算域での} A \text{ の逆元})$$

により与えられる。

【0059】後述する本実施形態に係る除算装置は、入力A、Bと法pに対して、 $B \cdot A' \cdot R \pmod{p}$ を効率的に求める装置である。

【0060】図13(b)に、p=23、R=2⁵ (n=5)とした場合の剩余系Z_pにおける元aと逆元xの対応関係を一例として示す。また、図13(c)に、p=23、R=2⁵ (n=5)とした場合のモンゴメリ演算域における元Aと逆元Xの対応関係を一例として示す。

【0061】さて、このようなモンゴメリ演算を利用するには、最初に剩余系Z_pからモンゴメリ演算域への変換を行ない、モンゴメリ演算域で所定の演算処理を繰り返した後、モンゴメリ演算域から剩余系Z_pへの逆変換を行なう。なお、これら変換・逆変換はそれぞれ多倍長乗算1回程度の処理であり、全体としてはあまり大きなオーバヘッドではない。

【0062】ここで、モンゴメリ演算を利用した具体例を示す。

【0063】例えば、p=23、R=2⁵ (n=5)として、モンゴメリ演算を利用し剩余系Z_pの元a=3の*

$$\begin{aligned} Y &= B \cdot (A^{-1} \cdot R^2) \cdot R^{-1} \pmod{p} \\ &= (B \cdot (A^{-1} \cdot R^2 \pmod{p}) \cdot R^{-1} \pmod{p}) \end{aligned}$$

と変形して、まず、除数Aと法pを入力としてモンゴメリ逆元計算部201によりモンゴメリ演算域でのAの逆元Xを求め、次に、このXと上記の被除数Bと法pを入力としてモンゴメリ乗算部202により $Y = B \cdot X \cdot R^{-1} \pmod{p}$ 、すなわち $Y = B \cdot A^{-1} \cdot R \pmod{p}$ を求める。

【0069】モンゴメリ乗算部202は、剩余系での乗算部に比べて高速に実現できることが知られており、本実施形態では例えば文献(1)などに開示された公知の

* 逆元xを求める場合を考える。まず、剩余系Z_pの元a=3をモンゴメリ演算域の元Aに変換すると、A=4が求まる。次に、モンゴメリ演算域の元A=4の逆元Xとして、X=3が求まる。そして、モンゴメリ演算域の元X=3を剩余系Z_pの元xに逆変換すると、x=8が求まる。なお、剩余系Z_pの元a=3の逆元xを直接、剩余系で求めると、x=8となり、上記の結果と一致することがわかる。

【0064】また、例えば、p=23、R=2⁵ (n=5)として、モンゴメリ演算を利用し剩余系Z_pの乗算3×6を行なう場合を考える。まず、剩余系Z_pの3と6をモンゴメリ演算域の4と8に変換し、 $Y = 4 \times 8 \times R^{-1} \pmod{p}$ から、Y=1が求まる。これを剩余系Z_pに逆変換すると、乗算結果としてy=18が求まる。なお、剩余系Z_pで直接 $y = 3 \times 6 \pmod{p}$ を求めると、y=18となり、上記の結果と一致することがわかる。

【0065】また、例えば、p=23、R=2⁵ (n=5)として、モンゴメリ演算を利用し剩余系Z_pの剩余除算6/2を行なう場合を考える。まず、剩余系Z_pの6と2をモンゴメリ演算域の8と18に変換し、被除数18の逆元として16を求め、 $Y = 8 / 18 \times R^{-1} \pmod{p} = 8 \times 16 \times R^{-1} \pmod{p}$ から、Y=4が求まる。これを剩余系Z_pに逆変換すると、除算結果としてy=3が求まる。なお、剩余系Z_pで直接 $y = 6 / 2 \pmod{p}$ を求めると、y=3となり、上記の結果と一致することがわかる。

【0066】以下では、本実施形態に係るモンゴメリ演算域での逆元計算装置および除算装置について説明する。

【0067】図1に、本発明の一実施形態に係るモンゴメリ演算域での除算装置の基本構成を示す。本除算装置200は、モンゴメリ逆元計算部201とモンゴメリ乗算部202を備えており、モンゴメリ逆元計算部201とモンゴメリ乗算部202をシーケンシャルに利用してモンゴメリ除算装置200を構成している。

【0068】本実施形態では、モンゴメリ除算 $Y = B \cdot A^{-1} \cdot R \pmod{p}$ を、

$$\begin{aligned} Y &= B \cdot (A^{-1} \cdot R^2) \cdot R^{-1} \pmod{p} \\ &= (B \cdot (A^{-1} \cdot R^2 \pmod{p}) \cdot R^{-1} \pmod{p}) \end{aligned}$$

技術を用いることができる。

【0070】図14に、モンゴメリ乗算部202における処理手順の一例を示す。ここでは、2数A、Bに対して、 $A \cdot B \cdot R^{-1} \pmod{N}$ を求めるものとして表記する。

【0071】V = -N⁻¹ mod R (すなわち、V·N = -1 mod Rを満たすV)を求める(ステップS101)、T = A · Bを求める(ステップS102)、W = ((T mod R) · V) mod Rを求める(ス

ステップS103)、 $T + W \cdot N$ をTに代入し(ステップS104)、 $T = T / R$ を実行する(ステップS105)。そして、 $T > N$ ならば(ステップS106)、TからNを減ずる(ステップS107)。このときに得られるTが法をNとするAとBのモンゴメリ乗算結果である。なお、Rを2のべき乗にとれば、剩余算や除算は2進数の下位の切り出しや上位の切り出しで実現できる。また、Vの値を法Nについて計算しておくことで、処理の効率化が可能である。

【0072】図15に、モンゴメリ乗算部202における処理手順の他の例を示す。この処理手順は図14の処理手順を改良し、必要な演算を多倍長乗算2回程度にしたものである。ここでは、2数A、Bに対して、 $A \cdot B \cdot R^{-1} \bmod N$ を求めるものとして表記する。また、A、B、Nなどは、基底bで表現されているものとする。例えば、 $A = a_{r-1} b^{r-1} + a_{r-2} b^{r-2} + \dots + a_1 b + a_0$ などの形式である。ここで、基底bは2のべき乗とし、例えば 2^8 、 2^{16} である。もちろん、b=2でも構わない。

【0073】 $v_0 = -N_0^{-1} \bmod b$ (すなわち、 $v_0 \cdot N_0 = -1 \bmod b$ を満たす v_0)を求め、また、 $T = 0$ 、 $i = 0$ とし(ステップS201)、 $T + a_i B b^i$ をTに代入し(ステップS202)、 $m_i = t_i v_0 \bmod b$ を求め(ステップS203)、 $T + m_i N b^i$ をTに代入し(ステップS204)、iを1増加する。次に、ステップS206で、 $i \leq (r-1)$ ならばステップS202に戻る。また、ステップS206で、 $i > (r-1)$ ならばループを抜けステップS207に移り、 $T = T / R$ を実行する(ステップS207)。そして、 $T > N$ ならば(ステップS208)、TからNを減ずる(ステップS209)。このときに得られるTが法をNとするAとBのモンゴメリ乗算結果である。

【0074】この手順の1つの特徴は、ステップS207の実行直前において、Tの下位側のLブロックが全て0(すなわち、TはRの倍数)になる点にある。したがって、ワーク領域を削減することが可能となる。また、剩余算は2進数の下位の切り出しで実現できる。また、 v_0 の値を法Nについて計算しておくことで、処理の効率化が可能である。

【0075】以上のようにモンゴメリ乗算部202は、効率的な実現が可能である。また、詳しくは後述するが、本発明によればモンゴメリ逆元計算部201を効率的に実現することができる。従って、本実施形態の除算装置200によれば、モンゴメリ演算域での除算を効率良く実行することができる。

【0076】図2に、本実施形態に係るモンゴメリ演算域での逆元計算装置201の基本構成の一例を示す。もちろん、この逆元計算装置201は、図1のモンゴメリ除算装置200のモンゴメリ逆元計算部201として用

いることができる。

【0077】本逆元計算装置201は、逆元計算部301と逆元補正部302を備えており、逆元計算部301と逆元補正部302をシーケンシャルに利用してモンゴメリ逆元計算装置201を構成している。

【0078】本実施形態では、逆元計算 $X = A^{-1} \cdot R^2 \bmod p = A^{-1} \cdot 2^{2n} \bmod p$ を、

$$X = A^{-1} \cdot (2^k \cdot 2^{2n-k}) \bmod p$$

$$= (A^{-1} \cdot 2^k) \cdot 2^{2n-k} \bmod p$$

$$= (A^{-1} \cdot 2^k \bmod p) \cdot 2^{2n-k} \bmod p$$

$$= C \cdot 2^{2n-k} \bmod p$$

と変形して、まず、整数Aと法p(ただしAはpと互いに素)を入力として逆元計算部301により $C = A^{-1} \cdot 2^k \bmod p$ とkを求める。ここで、kはL以上2L以下の整数で、Aとpから一意に決定される値である。次に、このCとkと法pを入力として逆元補正部302により逆元 $X = C \cdot 2^{2n-k} \bmod p$ 、すなわち $X = A^{-1} \cdot R^2 \bmod p$ を求める。

【0079】図3および図4に、逆元計算部301における処理手順の一例を示す。

【0080】この手順は、Uレジスタ、Vレジスタ、Tレジスタ、Sレジスタの4つの多倍長レジスタを利用して、レジスタの左右へのシフト演算とレジスタ同士の加算、減算で構成され、ループの繰り返し回数がループカウンタとして用いる変数kに格納される。kの値は法pのビットサイズをlとするとき、L以上2L以下であり、加減算の処理量はO(L)であるため、全体でも高々O(L²)の処理量である。以下、図3および図4の手順を流れを追って説明する。

【0081】まず、与えられた法pをレジスタUに、p以下の正の整数AをレジスタVに設定する。また、レジスタTに0、レジスタSに1、レジスタkに0をそれぞれ設定する(ステップS401)。以上が、変数初期化の処理である。

【0082】以降、ステップS402からステップS410の処理を、レジスタVが正の値である間(0になるまで)、繰り返す。

【0083】まず、レジスタVが0でなければ、繰り返し処理を続けるので、ステップS403にとぶ(ステップS402)。

【0084】レジスタUの最下位ビットが0か否かを判定する(ステップS403)。もし0であれば、レジスタUを右に1ビットシフトし(ステップS411)、レジスタSを左に1ビットシフトして(ステップS412)、ステップS410にとぶ。そして、ステップS410にてkの値を1増加し、ステップS402に戻る。

【0085】ステップS403にてレジスタUの最下位ビットが0でなければ、レジスタVの最下位ビットが0か否かを判定する(ステップS404)。もし0であれば、レジスタVを右に1ビットシフトし(ステップS4

13)、レジスタTを左に1ビットシフトして(ステップS414)、ステップS410にとぶ。そして、ステップS410にてkの値を1増加し、ステップS402に戻る。

【0086】ステップS403にてレジスタUの最下位ビットが0でなく、ステップS404にてレジスタVの最下位ビットが0でなければ、レジスタUとVの大小比較を行なう(ステップS405)。

【0087】もしU>Vならば、レジスタUからレジスタVの内容を引き(ステップS415)、レジスタUを右に1ビットシフトし(ステップS416)、レジスタTにレジスタSの内容を加算し(ステップS417)、レジスタSを左に1ビットシフトする(ステップS418)。そして、ステップS410にてkの値を1増加し、ステップS402に戻る。

【0088】もしステップS405の結果、U<VもしくはU=Vの場合は、レジスタVからレジスタUの内容を引き(ステップS406)、レジスタVを右に1ビットシフトし(ステップS407)、レジスタSにレジスタTの内容を加算し(ステップS408)、レジスタTを左に1ビットシフトする(ステップS409)。そして、ステップS410にてkの値を1増加し、ステップS402に戻る。

【0089】以上のループを繰り返し、ステップS402にてレジスタVが0になった場合、ステップS419に移る。そして、まず、レジスタUの内容が1かどうかをチェックする。レジスタUの内容は入力Aと法pの最大公約数になるので、もしUが1でなければAとpは互いに素でないことになるので、Aの逆元は存在しない。そのため、ステップS423でエラー処理をして終了する。

【0090】エラーでない場合、すなわちステップS419にてレジスタUの内容が1である場合、ステップS420でTとpの大小比較を行ない、もしTがp以上であれば、Tからpを引き(ステップS421)、Tがp以下の整数になるようにする。そして、ステップS422でpからTの内容を引いた結果をTに格納して処理を終了する。

【0091】以上の処理により、Tの内容には $A^{-1} \cdot 2^k \bmod p$ の計算結果が格納される。

【0092】次に、この $T = A^{-1} \cdot 2^k \bmod p$ とkの値を逆元補正部302に入力して、モンゴメリ逆元値を計算する。

【0093】図5に、逆元補正部302の処理手順の一例を示す。以下、図5の手順を流れを追って説明する。

【0094】まず、 $R = 2^n$ であるところのnを2倍した値をLにし、ループカウンタiを0に設定する(ステップS501)。

【0095】次に、ループの繰り返し回数としてL-kの値を求め、mに設定する(ステップS502)。

【0096】以降、ステップS503からステップS507の処理を、ループカウンタがmになるまで繰り返す。

【0097】すなわち、ステップS503でiとmを比較し、iがm未満の場合には、まず、レジスタTを1ビット左シフトする(ステップS504)。次に、Tとpの大小比較を行ない(ステップS505)、もしTがp以上であればTからpを引く(ステップS506)。そして、ステップS507にてiの値を1増加し、ステップS503に戻る。

【0098】ステップS503でi=mの場合には、上記の処理ループを抜ける。このループを抜けた時点でのTの値がモンゴメリ逆元値である。最後に、ステップS508で逆元の値Tを出力して処理を終了する。

【0099】以上に示した図3、図4、図5の手順はレジスタの加減算とビットシフトのみで実現されるため、効率的な装置化が可能である。

【0100】以下では、具体例を用いて、本実施形態のモンゴメリ逆元計算装置(または除算装置200のモンゴメリ逆元計算部)201の動作を説明する。

【0101】ここでは、一例として、法の値p=23、R=2ⁿ(n=5)としたときに、モンゴメリ演算域での元A=19の逆元を求める場合について示す。

【0102】図6(a)、(b)には、逆元計算部301における、Uレジスタ、Vレジスタ、Tレジスタ、Sレジスタのそれぞれの内容を2進数で表した値(ただし、TレジスタとSレジスタにおける上位側ビットの0の表示は省略してある)と、ループカウンタkの内容を10進数で表した値の変遷を、各処理ループについて示す。

【0103】また、図6(c)には、逆元補正部302における、Tレジスタの内容を2進数で表した値(ただし、上位側ビットの0の表示は省略してある)の変遷を、各処理ループについて示す。

【0104】まず、逆元計算部301において、初期化処理の結果、Uレジスタ=p=10111、Vレジスタ=A=10011、Tレジスタ=0、Sレジスタ=1、k=0が設定される。

【0105】1回目のループでは、U>VからステップS405にてYesとなり、ステップS415～S418とS410が実行される。この結果、Uレジスタ=00010、Vレジスタ=10011、Tレジスタ=1、Sレジスタ=10、k=1となる。

【0106】2回目のループでは、LSB(U)=0からステップS403にてYesとなり、ステップS411とS412とS410が実行される。この結果、Uレジスタ=00001、Vレジスタ=10011、Tレジスタ=10、Sレジスタ=100、k=2となる。

【0107】3回目のループでは、U<VからステップS405にてNoとなり、ステップS406～S409

19

とS410が実行される。この結果、Uレジスタ=0001、Vレジスタ=01001、Tレジスタ=10、Sレジスタ=101、k=3となる。

【0108】以上のようにして処理を繰り返した結果、7回目のループの実行後、Uレジスタ=00001、Vレジスタ=00000、Tレジスタ=100000、Sレジスタ=10111、k=7となり、Vレジスタ=0000となつたので、処理ループを抜ける。

【0109】次に、U=1であるからステップS419にてyesとなり、T>pであるからステップS420にYesとなり、この結果、ステップS421にてT=T-P=100000-10111=1001となり、最後にステップS422にてT=p-T=10111-1001=1110となる。

【0110】従つて、逆元計算部301の出力は、T=1110 (=10進数表現で14)、k=7となる。

【0111】次に、逆元補正部302において、初期状態として、T=1110、i=0、m=3に設定される(iとmの値は10進数で表す)。

【0112】1回目のループでは、Tレジスタが左シフトされてT=11100となり、ステップS505でYesとなるためステップS506にてT=101となり、そしてi=1となる。

【0113】2回目のループでは、Tレジスタが左シフトされてT=1010となり、ステップS505でNoとなるためステップS506は実行されず、i=2となる。

【0114】3回目のループでは、Tレジスタが左シフトされてT=10100となり、ステップS505でNoとなるためステップS506は実行されず、i=3=mとなり、処理ループを抜ける。

【0115】この結果、出力T=10100=10進数表現で20となる。

【0116】このようにして、法の値p=23、R=2ⁿ(n=5)としたときにおける、モンゴメリ演算域での元A=19の逆元X=20を得ることができる。

【0117】なお、モンゴメリ演算域での元19と20にそれぞれ対応する、剩余系Z_pでも元を求める、20と15になる。すなわち、剩余系Z_pでの元20の逆元として、15が得られたことになる。

【0118】以下では、前述した逆元計算部301や逆元補正部302における処理の他の例をそれぞれ示す。

【0119】まず、図7および図8に逆元計算部301の処理手順の他の例を示す。この手順は図3および図4に示した処理手順と原理的には同じであるが、図3および図4の手順においてステップS403とステップS404でそれぞれ多倍長レジスタUとVの最下位ビットだけを判定の基準に用い、最下位ビットが0の場合に後続の手順であるステップS411およびステップS412、ステップS413およびステップS414で1ビッ

10

トだけレジスタのシフトを行っていたものを、最下位ビットから0が複数連続する場合には一度に複数ビットのシフトを可能とするよう改良したものである。このように、一度に複数ビットをまとめて処理する方が有利な場合は多く、特にソフトウェア実装において高速となる。

【0120】以下、図7および図8の手順を流れを追って説明する。

【0121】まず、入力として与えられた法pをレジスタUに、p以下の正の整数AをレジスタVに設定する。また、レジスタTに0、レジスタSに1、レジスタkに0をそれぞれ設定する(ステップS601)。

【0122】以降、ステップS602からステップS612の処理を、レジスタVが正の値である間(0になるまで)、繰り返す。

【0123】ステップS602でレジスタVが0でなければ、まず、レジスタUの最下位ビットからの“0”的連長をカウントしてこの値をwとする(ステップS603)。次にwが0か否かを判定し(ステップS604)、もし0でなければ、レジスタUをwビットだけ右シフトし(ステップS613)、レジスタSをwビットだけ左シフトし(ステップS614)、ループカウンタKにwを加え(ステップS615)、ステップS602へとぶ。

【0124】ステップS604でwが0ならば、レジスタVの最下位ビットからの“0”的連長をカウントしてこの値をwとする(ステップS605)。次にwが0かどうかを判定し(ステップS606)、もし0でなければ、レジスタVをwビットだけ右シフトし(ステップS616)、レジスタTをwビットだけ左シフトし(ステップS617)、ループカウンタKにwを加え(ステップS618)、ステップS602へとぶ。

【0125】ステップS606でwが0ならば、レジスタUとVの大小比較をし(ステップS607)、もしU>Vならば、レジスタUからレジスタVの内容を引き(ステップS619)、レジスタUを右に1ビットシフトし(ステップS620)、レジスタTにレジスタSの内容を加算し(ステップS621)、レジスタSを左に1ビットシフトし(ステップS622)、ループカウンタKに1を加え(ステップS623)、ステップS602へとぶ。

【0126】もしステップS607の結果、U<VもしくはU=Vの場合は、レジスタVからレジスタUの内容を引き(ステップS608)、レジスタVを右に1ビットシフトし(ステップS609)、レジスタSにレジスタTの内容を加算し(ステップS610)、レジスタTを左に1ビットシフトし(ステップS611)、ループカウンタKに1を加え(ステップS623)、ステップS602へもどる。

【0127】以上のループを繰り返し、ステップS60

50

21

2でレジスタVが0になった場合、処理ループを抜け、まず、レジスタUの内容が1かどうかをチェックする（ステップS624）。レジスタUの内容は入力Aと法pの最大公約数になるので、もしUが1でなければステップS628でエラー処理をして終了する。

【0128】エラーでない場合、すなわちステップS624にてレジスタUの内容が1である場合、レジスタTとpの大小比較をし（ステップS625）、もしTがp以上であれば、Tからpを引き（ステップS626）、Tがp以下の整数になるようにする。そして、ステップS627でpからTの内容を引いた結果をTに格納して処理を終了する。

【0129】以上の処理により、Tの内容として $A^{-1} \cdot 2^k \bmod p$ が計算される。

【0130】次に、逆元補正部302の処理の流れの一例を図9、図10に示す。

【0131】この手順も図5に示した処理手順と原理的には同じであるが、図5の手順においてステップS504で多倍長レジスタTを1ビットだけ左シフトを行うことを繰り返していたものを、一度に複数ビットのシフトを可能とするように改良したものである。このように、一度に複数ビットをまとめて処理する方が有利な場合は多く、特にソフトウェア実装において高速となる。

【0132】以下、図9、図10の手順を流れを追って説明する。

【0133】まず、 $R = 2^n$ であるところのnを2倍した値をLにし、ループカウンタiを0に設定する（ステップS701）。

【0134】次に、ループの繰り返し回数としてL-kの値を求め、mに設定する（ステップS702）。

【0135】以降、ステップS703からステップS710の処理を、ループカウンタiがmになるまで繰り返す。

【0136】ステップS703でiとmを比較し、iがm未満の場合には、まず、レジスタTを法pと同じサイズの2進数と見たときに最上位ビットからの“0”的連長をカウントしてこの値をwとする（ステップS704）。

【0137】次に、wが0かどうかを判定し（ステップS705）、もし0ならばレジスタTを1ビットだけ左シフトする（ステップS712）。この結果、Tの値はpより大きくなるのでステップS713でレジスタTからpの値を減ずる。そして、ループカウンタiに1を加え（ステップS714）、ステップS703へとぶ。ステップS705でwが0でなければ、次にi+wを計算*

$$\begin{aligned}
 Y &= B \cdot A^{-1} \cdot R \bmod p \\
 &= B \cdot (A^{-1} 2^k) 2^{-k} \cdot (R^{-1} R^2) \bmod p \\
 &= B \cdot (A^{-1} 2^k \bmod p) \cdot R^{-1} (R^2 \cdot 2^{-k}) \bmod p \\
 &= (B \cdot C \cdot R^{-1} \bmod p) \cdot 2^{2n-k} \bmod p \\
 &= D \cdot 2^{2n-k} \bmod p
 \end{aligned}$$

22

*し、この値とmの大小比較を行う（ステップS706）。もし $i + w > m$ ならば、wを $m - i$ とし（ステップS715）、レジスタTをwビットだけ左シフトする（ステップS716）。ループカウンタはmとする（ステップS717）。ここではステップS716の左シフトの結果は必ずpよりも小さいため、補正は不要であり、ステップS703へとぶ。

【0138】ステップS706で $i + w < m$ もしくは $i + w = m$ の場合には、レジスタTをwビットだけ左シフトする（ステップS707）。この結果、Tの値はpより大きくなる可能性があるので、ステップS708でTとpの大小比較を行い、Tがp以上の場合はTからpの値を減ずる（ステップS709）。最後にループカウンタiにwを加え（ステップS710）、ステップS703に戻る。

【0139】以上のループを繰り返し、ステップS703で $i = m$ になった場合、処理ループを抜ける。このループを抜けた時点でのTの値がモンゴメリ逆元値であり、この値を出力して（ステップS711）、処理を終了する。

【0140】以上に示した図7と図8、図9と図10の手順もレジスタの加減算とビットシフトのみで実現される。

【0141】なお、図2のモンゴメリ逆元計算装置や図1のモンゴメリ除算装置のモンゴメリ逆元計算部において、逆元計算部301としては、図3と図4、図7および図8のいずれかを、また、逆元補正部302としては、図5、図9と図10のいずれかを、それぞれ任意に組み合わせて利用可能である。

【0142】ところで、本モンゴメリ除算装置は、図1で示した構成に限定されず、他の構成も可能である。図16に、本実施形態に係るモンゴメリ除算装置の基本構成の他の例を示す。

【0143】図2に例示したようにモンゴメリ逆元計算機装置は、一例として、逆元計算部301と逆元補正部302に分割できるが、図16では、この逆元計算部301をモンゴメリ乗算部202の前段に、また、逆元補正部302をモンゴメリ乗算部202の後段に配置した構成となっている。これは、逆元計算部301の出力Cに対しては、逆元補正部302の演算も、モンゴメリ乗算部202の演算も共に乗算であることから、その順序に対する可換性が成立することに基づいている。これを式で表現すると以下のようになる。

【0144】

このようにモンゴメリ除算装置は必ずしもモンゴメリ逆元計算部とモンゴメリ乗算部をシーケンシャルに利用しなくとも構成できる。重要なことは、モンゴメリ除算である $Y = B (A^{-1}) R \bmod p$ を効率的に計算することであり、そのためにモンゴメリ逆元計算装置の構成部品（モジュール）である逆元計算部と逆元補正部を分離して用いることもできる。

【0145】以上のように本実施形態によれば、橢円曲線暗号など多倍長の四則演算の繰り返し処理を高速に処理する場合に、モンゴメリ演算域での元を入力として、モンゴメリ逆元計算やモンゴメリ除算を実行することができる。したがって、最初に元の剩余系からモンゴメリ演算域に元を変換した後は、モンゴメリ演算域のままで繰り返し処理を実行できる。最後にモンゴメリ演算域から元の剩余系に逆変換すれば良いので全体としてのモンゴメリ変換・逆変換のオーバーヘッドを小さくできる。また、本実施形態のモンゴメリ逆元計算およびモンゴメリ除算は多倍長レジスタの加減算とビットシフトのみで実現できるため、ソフトウェア・ハードウェアのどちらでも効率良く実現できる。

【0146】以下では、本実施形態に係るモンゴメリ演算域での逆元計算装置のハードウェア構成について説明する。

【0147】図11に、本逆元計算装置の一構成例をブロック図で示す。

【0148】本逆元計算装置は、多倍長レジスタU(801)、多倍長レジスタV(802)、多倍長レジスタS(803)、多倍長レジスタT(804)とループカウンタとなる単精度のレジスタK(805)、演算部として加減算器806とビットシフタ807、加減算器806の出力を格納する多倍長のレジスタ808とビットシフタ807の出力を格納する多倍長のレジスタ809、そして全体の動作を制御する制御部(図示せず)を構成要素として持つ。これらの各構成要素は、データ・バス810に結線されており、相互にデータの転送が可能である。制御部では、前述したような処理手順に従いレジスタの特定ビットの0/1判定やレジスタの特定部分の0の連長の検査なども行う。

【0149】図2における逆元計算部301は、この構成要素を図3および図4、もしくは図7および図8に従う動作を行うように制御することによって実現され、図2における逆元補正部302は、法pを空いているレジスタ(例えばUレジスタ)に設定し、図5もしくは図9および図10に従う動作を行うように制御することによって実現される。

【0150】なお、以上の各機能は、ソフトウェアとしても実現可能である。

【0151】また、本実施形態は、コンピュータに所定の手順を実行させるための(あるいはコンピュータを所定の手段として機能させるための、あるいはコンピュー

タに所定の機能を実現させるための)プログラムを記録したコンピュータ読取り可能な記録媒体として実施することもできる。

【0152】本発明は、上述した実施の形態に限定されるものではなく、その技術的範囲において種々変形して実施することができる。

【0153】

【発明の効果】本発明によれば、モンゴメリ演算域での逆元計算と乗算を行って除算結果を得るので、モンゴメリ演算域の元を入力として、モンゴメリ演算域での除算結果を直接求めることができる。この結果、モンゴメリ演算域と元の剩余系との変換・逆変換のオーバーヘッドがないため、モンゴメリ演算域での除算が高速に実現できる。

【0154】また、本発明によれば、モンゴメリ演算域のままモンゴメリ逆元計算を行なうことができ、モンゴメリ演算域と元の剩余系との変換・逆変換のオーバーヘッドがないため、モンゴメリ演算域での逆元計算が高速に実現できる。

【0155】また、本発明によれば、モンゴメリ演算域での逆元計算が多倍長レジスタの加減算とビットシフトで実現できるため、ソフトウェア実装・ハードウェア実装のどちらでも高速な装置構成が可能となる。さらに、モンゴメリ演算域での除算も高速な装置構成が実現できる。

【0156】したがって、橢円曲線暗号などのように剩余系での乗算と除算を含む演算の繰り返し処理を基本演算とする暗号において、全体としての処理時間を高速化することができる。

【図面の簡単な説明】

【図1】本発明に係るモンゴメリ演算域での除算計算装置の一構成例を示す図

【図2】本発明に係るモンゴメリ演算域での逆元計算装置の一構成例を示す図

【図3】図2の逆元計算部での処理手順の一例を示すフローチャート

【図4】図2の逆元計算部での処理手順の一例を示すフローチャート

【図5】図2の逆元補正部での処理手順の一例を示すフローチャート

【図6】図2の逆元計算装置の具体的な動作例を説明するための図

【図7】図2の逆元計算部での処理手順の他の例を示すフローチャート

【図8】図2の逆元計算部での処理手順の他の例を示すフローチャート

【図9】図2の逆元補正部での処理手順の他の例を示すフローチャート

【図10】図2の逆元補正部での処理手順の他の例を示すフローチャート

【図11】本発明に係るモンゴメリ演算域での逆元計算装置のハードウェア構成を示すブロック図

【図12】モンゴメリ演算域と剰余系 Z_p の演算の対応を示す図

【図13】モンゴメリ演算域と剰余系 Z_p の元および逆元の具体例を示す図

【図14】図1のモンゴメリ乗算部での処理の一例を示すフローチャート

【図15】図1のモンゴメリ乗算部での処理の他の例を示すフローチャート

【図16】本発明に係るモンゴメリ演算域での除算計算装置の他の構成例を示す図

* 【符号の説明】

200…モンゴメリ除算装置

201…モンゴメリ逆元計算装置（モンゴメリ逆元計算部）

202…モンゴメリ乗算部

301…逆元計算部

302…逆元補正部

801, 802, 803, 804, 805, 808, 8

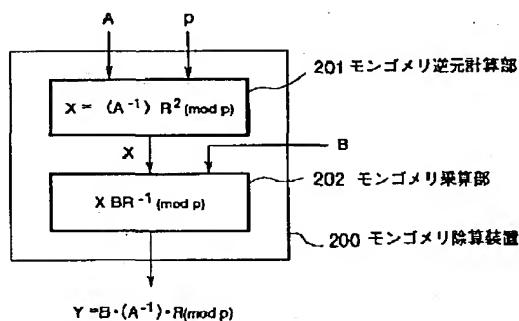
09…レジスタ

10 806…加減算器

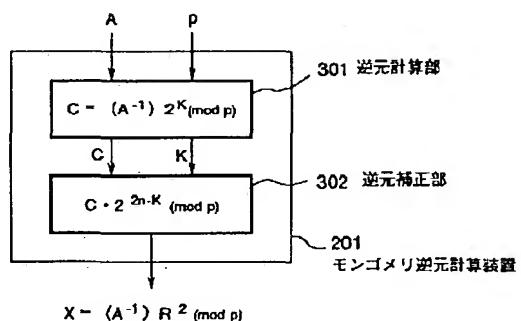
807…ビットシフタ

*

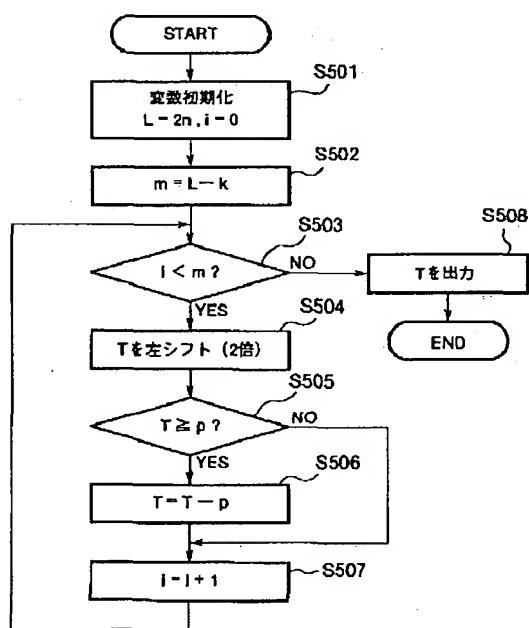
【図1】



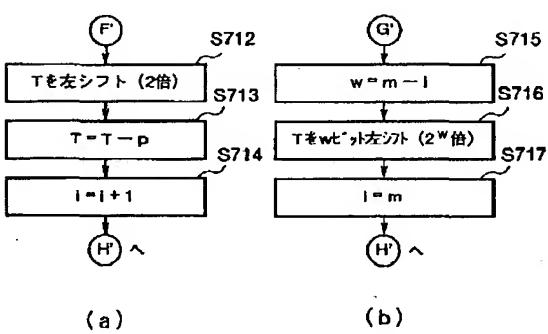
【図2】



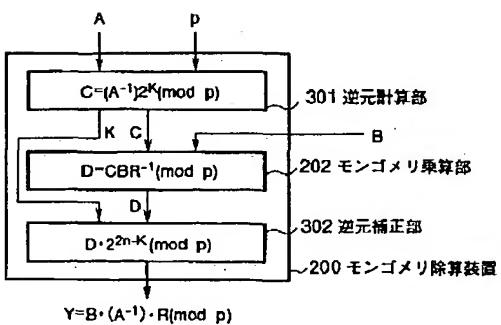
【図5】



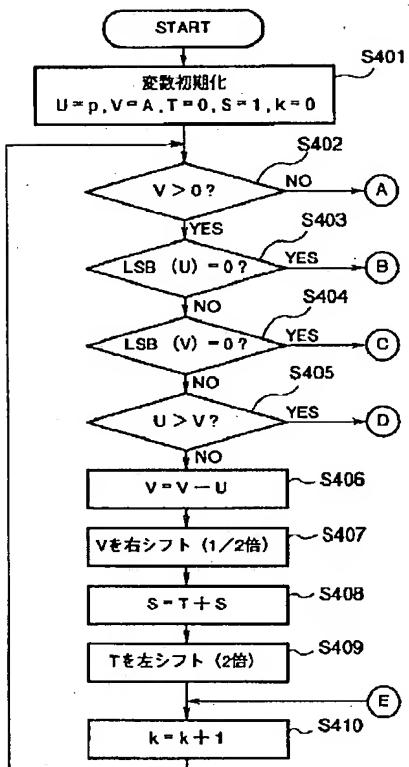
【図10】



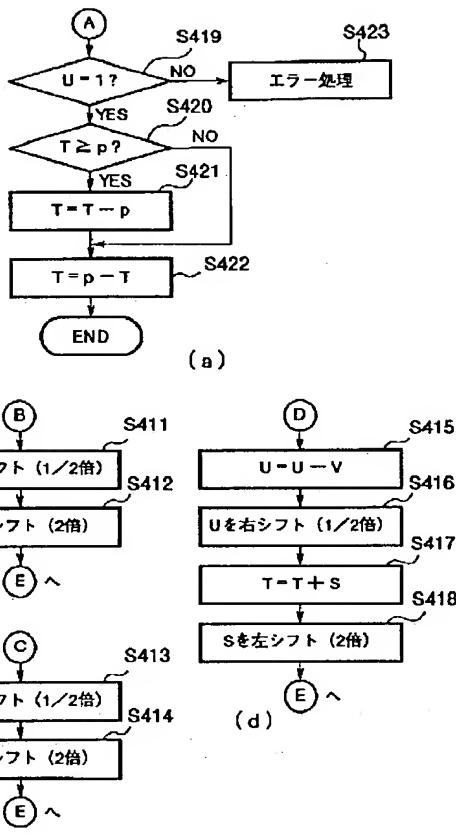
【図16】



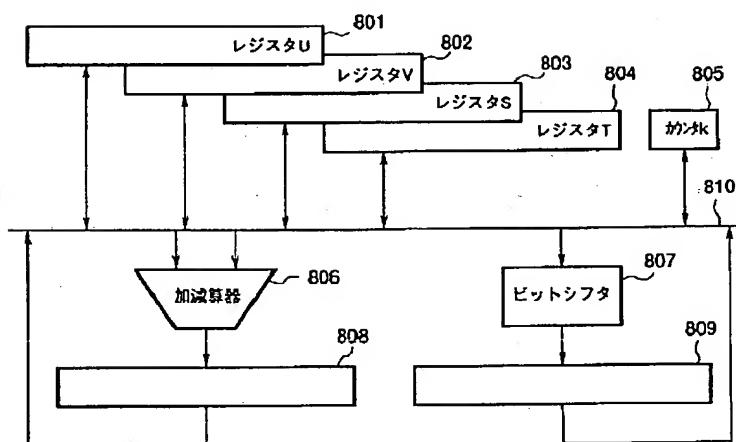
【図3】



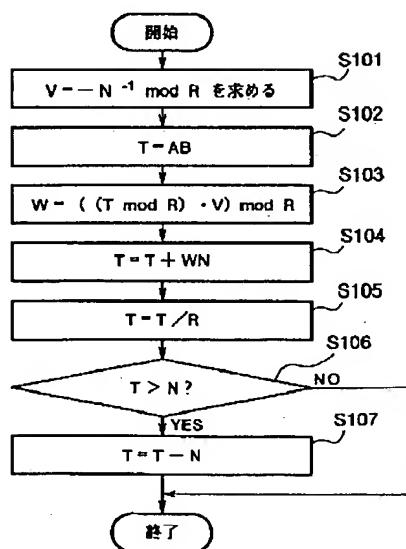
【図4】



【図11】



【図14】



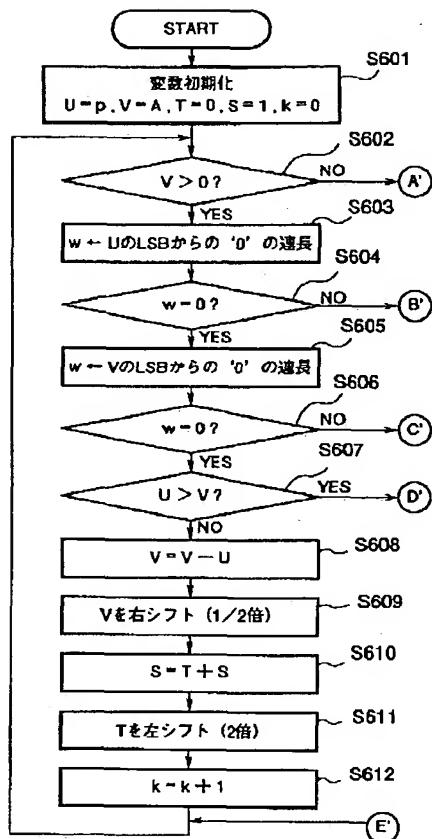
【図6】

	初期値	1回目	2回目	3回目	4回目
U ($\neg p$)	10111	00010	00001	00001	00001
V (=A)	10011	10011	10011	01001	00100
T	0	1	1	10	100
S	1	10	100	101	111
k	0	1	2	3	4

	5回目	6回目	7回目	出力値
U ($\neg p$)	00001	00001	00001	
V (=A)	00010	00001	00000	
T	1000	10000	100000	1110
S	111	111	10111	
k	5	8	7	7

	T
初期値	1110
1回目 左シフト	11100
pを減算	101
2回目 左シフト	1010
3回目 左シフト	10100

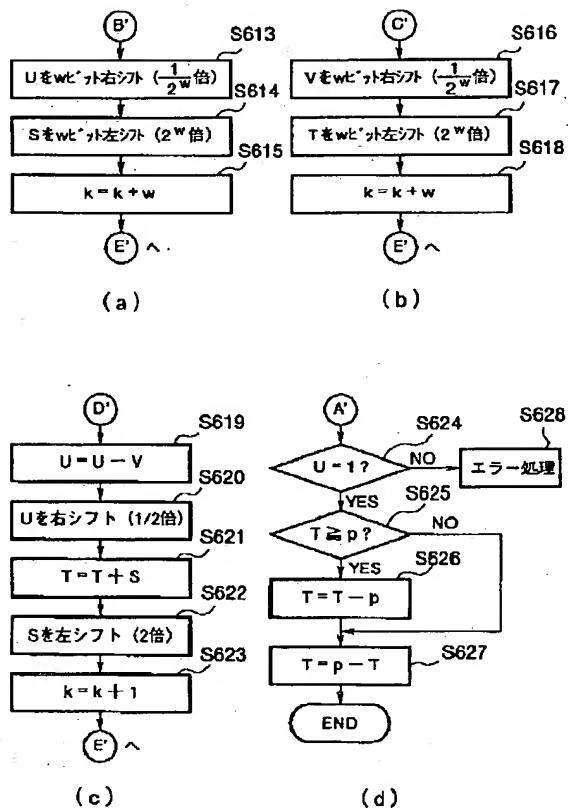
【図7】



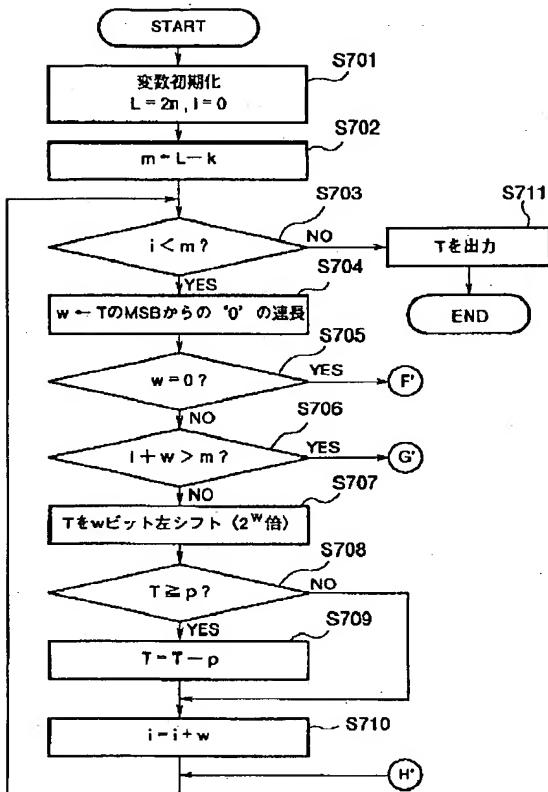
【図12】

	剰余系Zp領域	モンゴメリ演算域
定義域	0以上 $(p-1)$ 以下の整数	0以上 $(p-1)$ 以下の整数
元	$a = AR^{-1} \bmod p$	$A = aR \bmod p$
逆元	$ax = 1 \bmod p$ を満たすx	$Ax = R^2 \bmod p$ を満たすx
加算	$a + b \bmod p$	$A + B \bmod p$
減算	$a - b \bmod p$	$A - B \bmod p$
乗算	$ab \bmod p$	$ABR^{-1} \bmod p$
除算	$b/a = ba^{-1} \bmod p$ (ただし、 a^{-1} はaの逆元)	$B/A = BA^{-1}R^{-1} \bmod p$ (ただし、 A^{-1} はAの逆元)

【図8】



【図9】



【図13】

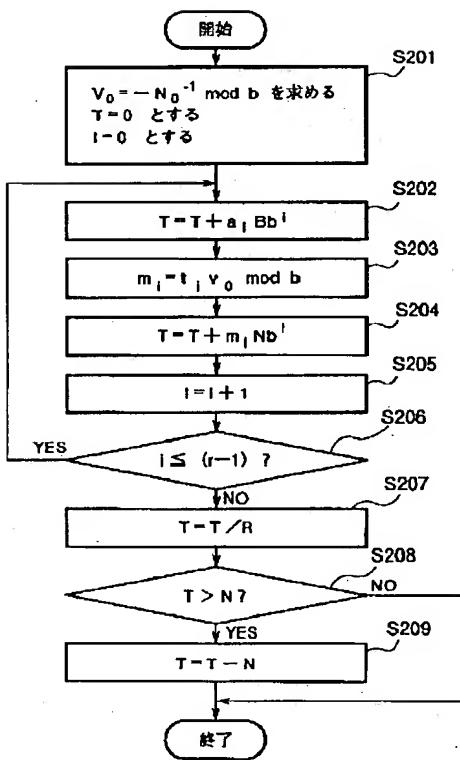
$A = a2^n \bmod p$ $p = 23 \quad (n=5)$		$ax = 1 \bmod p$ $p = 23 \quad (n=5)$		$AX = 2^{2n} \bmod p$ $p = 23 \quad (n=5)$	
a	A	a	x	A	X
0	0	1	1	1	12
1	9	2	12	2	6
2	18	3	8	3	4
3	4	4	6	4	3
4	13	5	14	5	7
5	22	6	4	6	2
6	8	7	10	7	5
7	17	8	3	8	13
8	3	9	18	9	9
9	12	10	7	10	15
10	21	11	21	11	22
11	7	12	2	12	1
12	16	13	16	13	8
13	2	14	5	14	14
14	11	15	20	15	10
15	20	16	13	16	18
16	6	17	19	17	21
17	15	18	9	18	16
18	1	19	17	19	20
19	10	20	15	20	19
20	19	21	11	21	17
21	5	22	22	22	11
22	14				

(a)

(b)

(c)

【図15】



【公報種別】特許法第17条の2の規定による補正の掲載
 【部門区分】第6部門第3区分
 【発行日】平成11年(1999)11月5日

【公開番号】特開平10-269060
 【公開日】平成10年(1998)10月9日
 【年通号数】公開特許公報10-2691
 【出願番号】特願平10-14250

【国際特許分類第6版】

G06F	7/72
G09C	1/00 650
// G06F	17/10
【F I】	
G06F	7/72
G09C	1/00 650 A
G06F	15/31 Z

【手続補正書】

【提出日】平成10年12月17日

【手続補正1】

【補正対象書類名】明細書

【補正対象項目名】請求項6

【補正方法】変更

【補正内容】

【請求項6】正の奇整数N、正の整数A（ $0 \leq A < N$ 、AとNは互いに素）について、Nを2進表現したときのビット長をlとして、 $n \geq l$ なる整数nに対して、 $X = A^{-1} \cdot 2^{2^n} \bmod N$ なるモンゴメリ演算域での逆元Xを求めるモンゴメリ逆元計算装置であって、初期状態を2進表現にてU=N、V=A、T=0、S=1、k=0とし、

Uの最下位ビットが0ならば、Uを右シフトし、Sを左シフトし、kを1增加する処理と、

Vの最下位ビットが0ならば、Vを右シフトし、Tを左シフトし、kを1増加する処理と、

Uの最下位ビットが1かつVの最下位ビットが1で、U>Vならば、UからVを減じ、Uを右シフトし、TにSを加え、Sを左シフトし、kを1増加する処理と、

Uの最下位ビットが1かつVの最下位ビットが1で、U≤Vならば、VからUを減じ、Vを右シフトし、SにTを加え、Tを左シフトし、kを1増加する処理からなる一連のループ処理を、V>0の間、繰り返し、

V=0になった場合、T≥NならばTからNを減じた後に、NからTを減じた値をTとし、T<NならばNからTを減じた値をTとする逆元計算手段と、

初期状態をi=0とし、

前記逆元計算手段により求められたTを左シフトした後、T≥NならばTからNを減じてiを1増加し、T<Nならばiを1増加するループ処理を、i<2n-kの間、繰り返し、

i=2n-kになったときのTを逆元Xとする逆元補正手段とを備えたことを特徴とするモンゴメリ逆元計算装置。

【手続補正2】

【補正対象書類名】明細書

【補正対象項目名】0025

【補正方法】変更

【補正内容】

【0025】本発明（請求項6）は、正の奇整数N、正の整数A（ $0 \leq A < N$ 、AとNは互いに素）について、Nを2進表現したときのビット長をlとして、 $n \geq l$ なる整数nに対して、 $X = X = A^{-1} \cdot 2^{2^n} \bmod N$ なるモンゴメリ演算域での逆元Xを求めるモンゴメリ逆元計算装置であって、初期状態を2進表現にてU=N、V=A、T=0、S=1、k=0とし、Uの最下位ビットが0ならば、Uを右シフトし、Sを左シフトし、kを1増加する処理と、Vの最下位ビットが0ならば、Vを右シフトし、Tを左シフトし、kを1増加する処理と、Uの最下位ビットが1かつVの最下位ビットが1で、U>Vならば、UからVを減じ、Uを右シフトし、TにSを加え、Sを左シフトし、kを1増加する処理と、Uの最下位ビットが1かつVの最下位ビットが1で、U≤Vならば、VからUを減じ、Vを右シフトし、SにTを加え、Tを左シフトし、kを1増加する処理からなる一連のループ処理を、V>0の間、繰り返し、V=0になった場合、T≥NならばTからNを減じた後に、NからTを減じた値をTとし、T<NならばNからTを減じた値をTとする逆元計算手段と、初期状態をi=0とし、前記逆元計算手段により求められたTを左シフトした後、T≥NならばTからNを減じてiを1増加し、T<Nならばiを1増加するループ処理を、i<2n-kの間、繰り返し、i=2n-kになったときのTを逆元Xとす

特開平10-269060

る逆元補正手段とを備えたことを特徴とする。